

# Package ‘neonUtilities’

August 3, 2019

**Version** 1.3.1

**Date** 2019-08-01

**Title** Utilities for Working with NEON Data

**Author**

Christine Laney <claney@battelleecology.org>, Claire Lunch <clunch@battelleecology.org>

**Maintainer** Claire Lunch <clunch@battelleecology.org>

**Description** NEON data packages can be accessed through the NEON Data Portal <<http://data.neonscience.org>> or through the NEON Data API (see <<http://data.neonscience.org/data-api>> for documentation). Data delivered from the Data Portal are provided as monthly zip files packaged within a parent zip file, while individual files can be accessed from the API. This package provides tools that aid in discovering, downloading, and reformatting data prior to use in analyses. This includes downloading data via the API, merging data tables by type, and converting formats. For more information, see the readme file at <<https://github.com/NEONScience/NEON-utilities>>.

**Depends** R (>= 3.4.0)

**biocViews**

**Imports** dplyr(>= 0.7.1), gdata(>= 2.18), httr, jsonlite, downloader, data.table, utils, stats, lubridate, readr, tidyr

**Suggests** rhdf5, testthat

**License** AGPL-3

**URL** <https://github.com/NEONScience/NEON-utilities>

**BugReports** <https://github.com/NEONScience/NEON-utilities/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-08-02 22:10:02 UTC

## R topics documented:

byFileAOP . . . . .	2
byTileAOP . . . . .	3
getAvg . . . . .	4
getDatatable . . . . .	5
getFileSize . . . . .	6
getPackage . . . . .	7
getProductInfo . . . . .	8
getProductSensors . . . . .	9
getTaxonTable . . . . .	9
getVarsEddy . . . . .	10
listFilesInZip . . . . .	11
listZipfiles . . . . .	12
loadByProduct . . . . .	12
stackByTable . . . . .	14
stackDataFiles . . . . .	15
stackEddy . . . . .	15
table_types . . . . .	16
transformFileToGeoCSV . . . . .	17
unzipZipfile . . . . .	18
zipsByProduct . . . . .	18
zipsByURI . . . . .	20
<b>Index</b>	<b>22</b>

---

byFileAOP

*Serially download all AOP files for a given site, year, and product*

---

### Description

Query the API for AOP data by site, year, and product, and download all files found, preserving original folder structure. Downloads serially to avoid overload; may take a very long time.

### Usage

```
byFileAOP(dpID, site = "SJER", year = "2017", check.size = TRUE,
  savepath = NA)
```

**Arguments**

dpID	The identifier of the NEON data product to pull, in the form DPL.PRNUM.REV, e.g. DP1.10023.001
site	The four-letter code of a single NEON site, e.g. 'CLBJ'.
year	The four-digit year to search for data. Defaults to 2017.
check.size	T or F, should the user be told the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.
savepath	The file path to download to. Defaults to NA, in which case the working directory is used.

**Value**

A folder in the working directory, containing all files meeting query criteria.

**Author(s)**

Claire Lunch <clunch@battelleecology.org> Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
## Not run:
# To download 2017 vegetation index data from San Joaquin Experimental Range:
byFileAOP(dpID="DP3.30026.001", site="SJER", year="2017")

## End(Not run)
```

---

byTileAOP	<i>Download AOP tiles overlapping specified coordinates for a given site, year, and product</i>
-----------	---

---

**Description**

Query the API for AOP data by site, year, product, and tile location, and download all files found. Downloads serially to avoid overload; may take a very long time.

**Usage**

```
byTileAOP(dpID, site = "SJER", year = "2017", easting, northing,
  buffer = 0, check.size = TRUE, savepath = NA)
```

**Arguments**

dpID	The identifier of the NEON data product to pull, in the form DPL.PRNUM.REV, e.g. DP1.10023.001
site	The four-letter code of a single NEON site, e.g. 'CLBJ'.
year	The four-digit year to search for data. Defaults to 2017.
easting	A vector containing the easting UTM coordinates of the locations to download.
northing	A vector containing the northing UTM coordinates of the locations to download.
buffer	Size, in meters, of the buffer to be included around the coordinates when determining which tiles to download. Defaults to 0. If easting and northing coordinates are the centroids of NEON TOS plots, use buffer=20.
check.size	T or F, should the user be told the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.
savepath	The file path to download to. Defaults to NA, in which case the working directory is used.

**Value**

A folder in the working directory, containing all files meeting query criteria.

**Author(s)**

Claire Lunch <clunch@battelleecology.org> Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

getAvg

*Get a list of the available averaging intervals for a data product*

---

**Description**

Most IS products are available at multiple averaging intervals; get a list of what's available for a given data product

**Usage**

getAvg(dpID)

**Arguments**

dpID	The identifier of the NEON data product, in the form DPL.PRNUM.REV, e.g. DP1.00006.001
------	--

**Value**

A vector of the available averaging intervals, typically in minutes.

**Author(s)**

Claire Lunch <clunch@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
# Get available averaging intervals for PAR data product
getAvg("DP1.00024.001")
```

---

getDatatable	<i>Get NEON data table</i>
--------------	----------------------------

---

**Description**

This is a function to retrieve a data table from the NEON data portal for sites and dates provided by the enduser. NOTE that this function only works for NEON Observation System (OS) data products, and only for select tables

**Usage**

```
getDatatable(dpid = NA, data_table_name = NA,
  sample_location_list = NA, sample_location_type = "siteID",
  sample_date_min = "2012-01-01", sample_date_max = Sys.Date(),
  sample_date_format = "%Y-%m-%d", data_package_type = "basic",
  url_prefix_data = "https://data.neonscience.org/api/v0/data/",
  url_prefix_products = "https://data.neonscience.org/api/v0/products/")
```

**Arguments**

dpid	character sting for NEON data product ID
data_table_name	character sting for name of the data table to download, e.g., 'sls_soilCoreCollection'
sample_location_list	list of sites, domains, etc. If NA, retrieve all data for the given data table / dpid combination.
sample_location_type	character sting for location type, such as 'siteID'. Must be one of the NEON controlled terms. If you're unsure, use 'siteID'

**sample\_date\_min**  
 start date for query. Default is 1-Jan-2012, and this should capture the earliest NEON data record.

**sample\_date\_max**  
 end date for query. Default is current date.

**sample\_date\_format**  
 date format. Default/expected format is yyyy-mm-dd

**data\_package\_type**  
 package type, either 'basic' or 'expanded'. If unsure, use 'expanded'

**url\_prefix\_data**  
 data endpoint for NEON API.

**url\_prefix\_products**  
 products endpoint for NEON API.

**Value**

data frame with selected NEON data

**Author(s)**

Eric R. Sokol <esokol@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```

sls_soilCoreCollection <- getDatatable(
  sample_location_list = c('CPER', 'TALL'),
  sample_date_min = '2014-01-01',
  sample_date_max = '2014-02-01',
  dpid = "DP1.10086.001",
  data_table_name = 'sls_soilCoreCollection')

```

---

getFileSize

*Get a file's size in megabytes*

---

**Description**

For any file, read the size and translate into a human readable string

**Usage**

```
getFileSize(filepath)
```

**Arguments**

filepath            The path to the file

**Value**

The size of the file in megabytes

**Author(s)**

Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

getPackage            *Get NEON data package*

---

**Description**

Get a zipped file for a single data product, site, and year-month combination. Use the NEON data portal or API to determine data availability by data product, site, and year-month combinations.

**Usage**

```
getPackage(dpID, site_code, year_month, package = "basic",  
          savepath = getwd())
```

**Arguments**

dpID                The identifier of the NEON data product to pull, in the form DPL.PRUNUM.REV, e.g. DP1.10023.001

site\_code           A four-letter NEON research site code, such as HEAL for Healy.

year\_month         The year and month of interest, in format YYYY-MM.

package            Either 'basic' or 'expanded', indicating which data package to download. Defaults to basic.

savepath           The location to save the output files to

**Value**

A zipped monthly file

**Author(s)**

Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

getProductInfo	<i>Get NEON data product information</i>
----------------	--

---

**Description**

Use the NEON API to get data product information such as availability, science team, etc.

**Usage**

```
getProductInfo(dpID)
```

**Arguments**

dpID	The data product id (optional), formatted as DP#.#####.###
------	--

**Value**

A data table with product information

**Author(s)**

Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
# Get documentation and availability of plant foliar properties data product  
cfcInfo <- getProductInfo("DP1.10026.001")
```



---

getProductSensors      *Get data product-sensor relationships*

---

**Description**

Pull all data from the NEON API /products endpoint, create a data frame with data product ID, data product name, and sensor type.

**Usage**

```
getProductSensors()
```

**Value**

A data frame

**Author(s)**

Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
## Not run:  
sensors <- getProductSensors()  
  
## End(Not run)
```

---

getTaxonTable      *Get NEON taxon table*

---

**Description**

This is a function to retrieve a taxon table from the NEON data portal for the taxon type by the enduser.

**Usage**

```
getTaxonTable(taxonType = NA, recordReturnLimit = NA,  
              stream = "true")
```

**Arguments**

taxonType            Character string for the taxonTypeCode. Must be one of ALGAE, BEETLE, BIRD, FISH, HERPETOLOGY, MACROINVERTEBRATE, MOSQUITO, MOSQUITO\_PATHOGENS, SMALL\_MAMMAL, PLANT, TICK

recordReturnLimit    Integer. The number of items to limit the result set to. If NA, will return all records in table.

stream                Character string, true or false. Option to obtain the result as a stream. Utilize for large requests.

**Value**

data frame with selected NEON data

**Author(s)**

Eric R. Sokol <esokol@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
# taxonTypeCode must be one of
# ALGAE, BEETLE, BIRD, FISH,
# HERPETOLOGY, MACROINVERTEBRATE, MOSQUITO, MOSQUITO_PATHOGENS,
# SMALL_MAMMAL, PLANT, TICK
#####
# return the first 4 fish records
taxa_table <- getTaxonTable('FISH', recordReturnLimit = 4)

# return all mammal taxa records
taxa_table <- getTaxonTable('SMALL_MAMMAL')
```

---

getVarsEddy

*Extract list of eddy covariance tables from HDF5 files*

---

**Description**

Extracts a list of table metadata from a single HDF5 file. Specific to eddy covariance data product: DP4.00200.001. Can inform inputs to stackEddy(); variables listed in 'name' are available inputs to the 'var' parameter in stackEddy().

**Usage**

```
getVarsEddy(filepath)
```

**Arguments**

filepath            The folder containing the H5 file [character]

**Value**

A data frame of the metadata for each data table in the HDF5 file

**Author(s)**

Claire Lunch <clunch@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
## Not run:
# read variables from a file in a hypothetical filepath
ec.vars <- getVarsEddy(filepath='/data/NEON.D19.BONA.DP4.00200.001.nsaе.2017-12.basic.h5')

## End(Not run)
```

---

listFilesInZip	<i>Get a data frame with the names of all files within a zipped NEON data package</i>
----------------	---

---

**Description**

Given the top level zip file, return dataframe of all of the files within it without unzipping the file

**Usage**

```
listFilesInZip(zippath)
```

**Arguments**

zippath            The path to a zip file

**Value**

A list of filenames within the given zip file

**Author(s)**

Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

listZipfiles	<i>Get all zip file names within a zipped NEON data package</i>
--------------	---

---

**Description**

Given the data frame of all the files within the top level zip file, return an array of just the zip file names (no pdf, xml, or other files).

**Usage**

```
listZipfiles(zippath)
```

**Arguments**

zippath	The path to a zip file
---------	------------------------

**Value**

An array of all zip files contained within the focal zip file

**Author(s)**

Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

loadByProduct	<i>Get files from NEON API, stack tables, and load into the current environment</i>
---------------	---

---

**Description**

Pull files from the NEON API, by data product, merge data for each table, and read into the current R environment

**Usage**

```
loadByProduct(dpID, site = "all", startdate = NA, enddate = NA,  
  package = "basic", avg = "all", check.size = TRUE)
```

**Arguments**

dpID	The identifier of the NEON data product to pull, in the form DPL.PRNUM.REV, e.g. DP1.10023.001
site	Either the string 'all', or the four-letter code of a single NEON site, e.g. 'CLBJ'. Future versions may allow more options for subsetting than one or all sites. Defaults to all.
startdate	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA.
enddate	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA.
package	Either 'basic' or 'expanded', indicating which data package to download. Defaults to basic.
avg	Either the string 'all', or the averaging interval to download, in minutes. Only applicable to sensor (IS) data. Defaults to 'all'.
check.size	T or F, should the user be told the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.

**Details**

All available data meeting the query criteria will be downloaded. Most data products are collected at only a subset of sites, and dates of collection vary. Consult the NEON data portal for sampling details. Dates are specified only to the month because NEON data are provided in monthly packages. Any month included in the search criteria will be included in the download. Start and end date are inclusive.

**Value**

A named list of all the data tables in the data product downloaded, plus a validation file and a variables file, as available.

**Author(s)**

Claire Lunch <clunch@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
## Not run:
# To download plant foliar properties data from all sites, expanded data package:
cfc <- loadByProduct(dpID="DP1.10026.001", site="all", package="expanded")

## End(Not run)
```

---

stackByTable                      *Join data files in a zipped NEON data package by table type*

---

### Description

Given a zipped data file, do a full join of all data files, grouped by table type. This should result in a small number of large files.

### Usage

```
stackByTable(filepath, savepath = NA, folder = FALSE,
             saveUnzippedFiles = FALSE, dpID = NA)
```

### Arguments

filepath	The location of the zip file
savepath	The location to save the output files to
folder	T or F: does the filepath point to a parent, unzipped folder, or a zip file? If F, assumes the filepath points to a zip file. Defaults to F.
saveUnzippedFiles	T or F: should the unzipped monthly data folders be retained?
dpID	Data product ID of product to stack. Not needed; defaults to NA, included for back compatibility

### Value

All files are unzipped and one file for each table type is created and written. If savepath="envt" is specified, output is a named list of tables; otherwise, function output is null and files are saved to the location specified.

### Author(s)

Christine Laney <claney@battelleecology.org> Claire Lunch <clunch@battelleecology.org>

### References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

### Examples

```
## Not run:
# To unzip and merge files downloaded from the NEON Data Portal
stackByTable("~/NEON_par.zip")

# To unzip and merge files downloaded using zipsByProduct()
stackByTable("~/filesToStack00024", folder=T)

## End(Not run)
```

---

stackDataFiles	<i>Join data files in a unzipped NEON data package by table type</i>
----------------	--

---

**Description**

Given a folder of unzipped files (unzipped NEON data file), do a full join of all data files, grouped by table type. This should result in a small number of large files.

**Usage**

```
stackDataFiles(folder)
```

**Arguments**

folder	The location of the data
--------	--------------------------

**Value**

One file for each table type is created and written.

**Author(s)**

Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

stackEddy	<i>Extract eddy covariance data from HDF5 format</i>
-----------	--

---

**Description**

Convert data of choice from HDF5 to tabular format. Specific to eddy covariance data product: DP4.00200.001

**Usage**

```
stackEddy(filepath, level = "dp04", var = NA, avg = NA)
```

**Arguments**

filepath	One of: a folder containing NEON EC H5 files, a zip file of DP4.00200.001 data downloaded from the NEON data portal, a folder of DP4.00200.001 data downloaded by the neonUtilities::zipsByProduct() function, or a single NEON EC H5 file [character]
level	The level of data to extract; one of dp01, dp02, dp03, dp04 [character]
var	The variable set to extract, e.g. co2Turb [character]
avg	The averaging interval to extract, in minutes [numeric]

**Details**

Given a filepath containing H5 files of DP4.00200.001 data, extracts variables, stacks data tables over time, and joins variables into a single table. For data product levels 2-4 (dp02, dp03, dp04), joins all available data, except for the flux footprint data in the expanded package. For dp01, an averaging interval and a set of variable names must be provided as inputs.

**Value**

A named list of data frames. One data frame per site, plus one data frame containing the metadata (objDesc) table and one data frame containing units for each variable (variables).

**Author(s)**

Claire Lunch <clunch@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
## Not run:
# To extract and merge Level 4 data tables, where data files are in the working directory
flux <- stackEddy(filepath=getwd(), level='dp04', var=NA, avg=NA)

## End(Not run)
```

---

table_types	<i>Publication table information</i>
-------------	--------------------------------------

---

**Description**

A dataset containing publication table names, descriptions, type (site-date, site-all, lab-all, lab-current), and a time index

**Usage**

```
table_types
```



**Format**

A data frame with 4 variables. Number of rows changes frequently as more tables are added:

**productID** data product ID

**tableName** name of table

**tableDesc** description of table

**tableType** type of table (important for knowing which tables to stack and which to not stack)

**tableTMI** a time index (e.g., 1 = native resolution or 1 minute, 30 = 30 minute averages or totals)

**Source**

dps database

---

transformFileToGeoCSV *Transform NEON CSV file to GeoCSV*

---

**Description**

Read in a NEON monthly data zip file and parse the respective variables file to create a new GeoCSV file

**Usage**

```
transformFileToGeoCSV(infile, varfile, outfile)
```

**Arguments**

infile	The path to the file that needs to be parsed
varfile	The path to the variables file needed to parse the infile
outfile	The path where the new GeoCSV file should be placed

**Value**

The same data file with a GeoCSV header

**Author(s)**

Christine Laney <claney@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

unzipZipfile	<i>Unzip a zip file either at just the top level or recursively through the file</i>
--------------	--

---

**Description**

Unzip a zip file either at just the top level or recursively through the file

**Usage**

```
unzipZipfile(zippath, outpath = substr(zippath, 1, nchar(zippath) - 4),
  level = "all")
```

**Arguments**

zippath	The filepath of the input file
outpath	The name of the folder to save unpacked files to
level	Whether the unzipping should occur only for the 'top' zip file, or unzip 'all' recursively, or only files 'in' the folder specified

**Author(s)**

Christine Laney <claney@battelleecology.org> Claire Lunch <clunch@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

zipsByProduct	<i>Get files from NEON API to feed the stackByTable() function</i>
---------------	--

---

**Description**

Pull files from the NEON API, by data product, in a structure that will allow them to be stacked by the stackByTable() function

**Usage**

```
zipsByProduct(dpID, site = "all", startdate = NA, enddate = NA,
  package = "basic", avg = "all", check.size = TRUE, savepath = NA,
  load = F)
```

**Arguments**

dpID	The identifier of the NEON data product to pull, in the form DPL.PRUNUM.REV, e.g. DP1.10023.001
site	Either the string 'all', meaning all available sites, or a character vector of 4-letter NEON site codes, e.g. c('ONAQ', 'RMNP'). Defaults to all.
startdate	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA.
enddate	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA.
package	Either 'basic' or 'expanded', indicating which data package to download. Defaults to basic.
avg	Either the string 'all', or the averaging interval to download, in minutes. Only applicable to sensor (IS) data. Defaults to 'all'.
check.size	T or F, should the user be told the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.
savepath	The location to save the output files to
load	T or F, are files saved locally or loaded directly? Used silently with loadByProduct(), do not set manually.

**Details**

All available data meeting the query criteria will be downloaded. Most data products are collected at only a subset of sites, and dates of collection vary. Consult the NEON data portal for sampling details. Dates are specified only to the month because NEON data are provided in monthly packages. Any month included in the search criteria will be included in the download. Start and end date are inclusive.

**Value**

A folder in the working directory (or in savepath, if specified), containing all zip files meeting query criteria.

**Author(s)**

Claire Lunch <clunch@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
## Not run:
# To download plant foliar properties data from all sites, expanded data package:
zipsByProduct(dpID="DP1.10026.001", site="all", package="expanded")

## End(Not run)
```

zipsByURI

*Get files from NEON ECS Bucket using URLs in stacked data***Description**

Read in a set of URLs from NEON data tables and then download the data from the NEON ECS buckets. Assumes data tables are in the format resulting from merging files using `stackByTable()`. File downloads from ECS can be extremely large; be prepared for long download times and large file storage.

**Usage**

```
zipsByURI(filepath, savepath = paste0(filepath, "/ECS_zipFiles"),
  pick.files = FALSE, check.size = TRUE, unzip = TRUE,
  saveZippedFiles = FALSE)
```

**Arguments**

filepath	The location of the NEON data containing URIs
savepath	The location to save the output files from the ECS bucket, optional. Defaults to creating a "ECS_zipFiles" folder in the filepath directory.
pick.files	T or F, should the user be told the name of each file before downloading? Defaults to F. When working in batch mode, or other non-interactive workflow, use pick.files=F.
check.size	T or F, should the user be told the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.
unzip	T or F, indicates if the downloaded zip files from ECS buckets should be unzipped into the same directory, defaults to T. Supports .zip and .tar.gz files currently.
saveZippedFiles	T or F: should the zip files be retained after unzipping? Defaults to F.

**Value**

A folder in the working directory (or in savepath, if specified), containing all files meeting query criteria.

**Author(s)**

Kaelin Cawley <kcawley@battelleecology.org>

**References**

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

**Examples**

```
## Not run:  
# To download stream morphology data from stacked data:  
zipsByURI(filepath="~/filesToStack00131/stackedFiles")  
  
## End(Not run)
```

# Index

## \*Topic **datasets**

table\_types, [16](#)

byFileAOP, [2](#)

byTileAOP, [3](#)

getAvg, [4](#)

getDatatable, [5](#)

getFilesize, [6](#)

getPackage, [7](#)

getProductInfo, [8](#)

getProductSensors, [9](#)

getTaxonTable, [9](#)

getVarsEddy, [10](#)

listFilesInZip, [11](#)

listZipfiles, [12](#)

loadByProduct, [12](#)

stackByTable, [14](#)

stackDataFiles, [15](#)

stackEddy, [15](#)

table\_types, [16](#)

transformFileToGeoCSV, [17](#)

unzipZipfile, [18](#)

zipsByProduct, [18](#)

zipsByURI, [20](#)